IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

An Effective and Private Approach Blockchain-based

1st DEEPTI SHARMA,2nd SHREYA DUNGARIA,3rd PREETI SAROJ

ABSTRACT: - A promising technology that increases automation and productivity for industrial applications is (IIoT). Devices from cooperating IIoT Sfields will interact and cooperate in more complicated industrial tasks. For securing cross-domain device interactions, we advise combining multi-factor authentication with the blockchain. Because multifactor authentication adheres to the operational modes of IIoT devices while simultaneously offering increased security levels, blockchain technology promotes confidence across multiple sectors. Unfortunately, this combined usage has drawbacks such as increased blockchain storage requirements, the risk of a loss of factor attack, and the tension between privacy protection and efficiency. We apply these findings in this research to develop a multi-factor device authentication mechanism for cross-domain IIoT that protects privacy. Specifically, before being converted into key materials, several parameters are also encoded by the hardware fingerprint into random numbers. Overhead is reduced by simply storing the dynamic accumulator for each domain, which collects derived key materials for devices. In addition, the cross-domain IIoT device unlinkable IDs can be successfully verified using the on-chain accumulator. The security of our protocol is explicitly demonstrated, and the qualities and functionality of the security are looked at one by one. The effectiveness and dependability of the system were demonstrated using an operating proof-of-concept prototype. On-chain storage has significantly decreased, according to comparative data.

I. INTRODUCTION

INTRODUCTION: - Thanks to recent advancements, the Industrial Internet of Things (IIoT) can now function as a machine-oriented platform in industrial environments. In order to build digital and smart manufacturing, machine-to-machine links are used to connect industrial assets (such as IIoT devices, resources, and systems). Information technology and operational procedures are coupled to provide flexible data gathering and sharing, on-demand command, remote real-time access, etc. IIoT devices from many collaborating domains can communicate information and operate together flexibly across the cloud-based network to efficiently finish the ever-more-complex manufacturing tasks. Thus, cross-domain collaboration in the IIoT has the potential to become a future industrial production paradigm and significantly boost productivity. In addition to malicious impersonation, physical cloning assaults could pose a hazard to IIoT equipment installed in public spaces. Because entities in each domain would only have faith in the domain administrator, it becomes challenging to build trust between many domains. Consequently, the significance of cross-domain device authentication in safeguarding cross-domain partnerships becomes evident. It may facilitate mutual trust and authentication across entities across different domains and offer a secure session key to protect the public channel. Multi-factor authentication schemes have been established recently to enhance the security

protection of IIoT systems [5]. These protocols integrate many components (biometrics, hardware, PIN, and password) from people and devices at the same time. This security technique fits in well with IIoT applications since completing industrial activities necessitates both autonomous device operation and human-device interaction [6]-[7]. However, it is challenging for a domain administrator to foster trust among various domains and to have complete confidence from all entities in other domains. Therefore, cross-domain IIoT scenarios cannot directly benefit from multi-factor authentication. A distributed and decentralized public digital ledger is essentially what the next blockchain technology is [8]. Without relying on a reliable centralized authority, data will be securely synchronized through transactions between peer nodes from various domains. The security measures in place [9]– [16] have made use of this property to foster trust between various domains.

In order to safeguard device collaboration and communications in cross-domain IIoT, the following issues still need to be fixed when utilizing blockchain and multi-factor authentication.

- The potential loss due to a factor assault: In order to prevent known attacks [17] on the multi-factor database at the server-side, it is recommended by contemporary research [5] and [6] to map factors to random integers and save them as public keys. However, the possible loss of factor attack present in IIoT contexts has not been taken into account by current multi-factor protection approaches. Without human input, devices typically need to do automatic production activities. In order to authenticate users, IIoT devices should separately offer sensitive factors . Given that attacker may obtain these device factors by doing power analysis on captured IIoT devices [18]. The malicious impersonation would then be carried out using these compromised factors
- Efficiency and protecting privacy are incompatible. The blockchain may often be used in one of two ways to create cross-domain authentication systems. Public keys are registered in advance by the first category in the blockchain ledger [10], [12]. By periodically querying the blockchain ledger, the server might obtain these keys. Another type of authentication transfers one-time pseudo-public keys as brand-new transactions [4], [11], and [14]. To maintain anonymity, this group exclusively encrypts genuine identities into public keys. Attackers can still link various messages to the device making the request as well as the public key being questioned. The second category's efficiency is constrained by the transaction throughput, but by providing a unique public key for every request, privacy is maintained to aid in unlinkability. Thus, it needs to think about how to strike a balance between the needs for maximum efficiency and privacy protection.
- Blockchain storage requirements: Each domain server should maintain a local key-value state database (like Level DB) for the logged transactions as well as a copy of the blockchain ledger in order to promote trust across domains [19]. Hence, as the quantity of on-chain public keys generated from diverse sources and domains rises, the domain server must have sufficient storage and RAM to manage the extra overhead. Off-chain storage was used in the study [4] to decrease the amount of data written to the blockchain. Additionally, the Merkle Patricia Tree and RSA Accumulator were adopted in work [20, 21] to improve the blockchain data structure. By altering the blockchain's traditional storage structure, this research cuts all overhead.

II. RELATED WORK

This section examines cross-domain security measures based on blockchain technology as well as multi-factor authentication protocols. Additionally, Table 1 provides an overview of the pertinent studies' specific contributions and unresolved common difficulties.

A. Factor protection and multi-factor authentication

This section looks into multi-factor authentication techniques with factor protection. These protocols enable stronger security levels by concurrently validating several factors (such a password, a biometric, and hardware) at the server-side. In order to protect the multi-factor database on the server and prevent attacks on the factors that were kept there, Li et al. [6] developed a multi-factor harden service utilizing the oblivious pseudo-random function. To further cut down on latency, a smooth protective hash function was incorporated into the low-interactivity authentication scheme. Similar to this, Zhang et al. proposed a similar paradigm to include several authentication factors in their work [5]. In order to attain secure storage, these elements will be

An overview of unresolved common issues and the specific roles played by current protocols

Group of Protocols	Unsolved common issues	Customs	Particular inputs
Multi-factor authentication with factors protection	1. Potential loss of factor attack 2. Trust among different domains	[5]	By addressing the arising security and efficiency challenges, their approach enhances single-factor authentication.
		[6]	It helps the standard model's low-interactivity safe multifactor protocol function.
		[22]	Pedersen commitment is used to provide remote multi-factor authentication that protects privacy.
Blockchain-based cross-domain authentication	3. Contradiction between efficiency and privacy preservation 4. On-chain storage overhead from protocol layer	[11]	It is built a dynamic key management architecture to transfer secret keys through blockchain transactions.
		[12]	In order to effectively handle cross-domain authentication, smart contracts are utilized in their work to manage system settings and public keys.
		[14]	An efficient way to maintain certificates is to combine the Block chain with a key generation technique.
		[4]	Blockchain technology is linked with identity-based cryptography to facilitate cross-domain device connections.
		[21]	Their method improves the data structure of blockchain by streamlining storage requirements through the application of the Merkle Patricia Tree.
Cross-domain authentication of combined usage	Address all above-mentioned issues	Our Work	We proposed combining multi-factor authentication with blockchain to safeguard cross-domain device interactions and collaboration.

turned into public aspects. A secure remote multifactor technique was created by Liu et al. [23] It reduces the communication cost and provides a reduced key size by utilizing the chaotic map. For these reasons, a privacy-preserving ZKPoK protocol was developed in the work [22]. The Pedersen commitment was made with the purpose of concurrently verifying biometrics and user passwords. Two fundamental problems often impede the integration of multi-factor authentication with cross-domain IIoT. The first problem is how to boost confidence between different areas. Second, a loss-of-factor attack would still be dangerous for the device-side even if these studies ensured the security of numerous factors at the server-side.

B. Cross-domain security mechanisms based on blockchain

Modern review papers [24] through [27] have concentrated on the marriage of blockchain and IoT. They described the fusion of these two technologies and their associated commercial uses. In addition, a number of significant issues and potential remedies were outlined. A notable study by Farrag et al. [25] Additionally, They completed the survey's assessment of security analysis techniques and consensus algorithms [27]. Guidelines for evaluating the effectiveness of blockchain-based IoT security and privacy solutions were also included in this article. Additionally, Kai et al. [26] examined potential research topics and security concerns associated with the application of smart contracts in the Internet of Things. These polls served as inspiration for using the blockchain to foster trust amongst various industriesRegarding devices and users of the IIoT that are cross-domain, an architecture based on blockchain known as xDBAuth. They conducted study and developed a hierarchy of local and global smart contracts to carry out permission delegation and access control, preventing unwanted delegations and safeguarding user privacy at the same time. Building cross-domain trust access techniques [28] for power terminals and boosting data credibility and efficacy for cross-domain IoT authentication are the goals of this project [15]. hierarchical structures have also been deployed. To enable signature transitivity across several domains, the digital signature and accumulator were integrated[29]. IoT Passport was developed as a trust framework for access control to allow cooperation across cross-platform devices[30]. On the blockchain, the signatures of cooperative device operations will be kept. A signature will be required to execute the permit, and a system of incentives will be developed. The work [31] provided blockchain-based hierarchical access management for privacy-conscious situations. Scalability was employed in the multiblock chain design to meet the requirements of low latency and high dependability. Identity-based cryptography was also included into the blockchain for cross-domain IoT in [4], [32], and [33]. By utilizing the blockchain, Jia et al. [32] were able to offer identity-based self-authentication in lieu of the traditional trusted certificate authority. The blockchain network needs to distribute the one-time identity-based public keys to several IIoT domains in order to offer unlikable cross-domain device authentication [4].Decentralized identity management was established by [33]. Furthermore, without disclosing the user's privacy, the identity information was sent via consensus methods to many domains.

III. PREliminary Discussions

A. A method of deriving multi-factor keys using hardware assistance PUFs,

- It can be compared to physical roots and hardware fingerprints and are created by the manufacturing variations of integrated circuits. PUFs act as a secure one-way function {0,1} n --{0,1} m in most cases. An unexpected and unclonable m-bit answer is produced in response to an n-bit challenge. In order to generate unique PUF seeds for different IIoT application operating modes, the physically secure PUF answer first combines with various random integers. Multiple factors will be encoded into random integers by the PUF seed. Next, random numbers are transformed into private and public key pairs using BIP32, an elliptic curve encryption-based system. The multi-factor database's security is guaranteed by storing only public keys on the server. Even if factors are made public, it is impossible to

precisely extract the private key without the PUF seed. This is also the foundation for the defense against a loss-of-factor attack on our protocol. By including a changeable and recoverable secret CI into the generation of the child private key ski,j, our method also removes the security concern of key leaking [14] that is included in BIP32.

- (ki,ci)=DerivePsk(R,ri,factors) is the process for creating a parent secret key, where R, ri, i, and factors stand for the PUF response. The list of variables, the operation mode, and the random number. PUF seed K(root,i) is first calculated by H1(R||ri). The working mode I and the input factors ft (0,1,2...n) are then encoded by the PUF seed into random numbers, which it then converts into parent secret keys (ki,ci). Remark: Every IIoT device may perform tasks requiring human-machine interaction or autonomous production by switching to a different operating mode.
- Child private key derivation: The procedure for determining a child's private key is defined as Cski,j=DeriveCsk(ki,ci,j). In elliptic curve cryptography, the components ci and ki are converted to Ci and Ki by scalar multiplication of the curve generator P. In the event that j is the number of the child private/public key pair, kiH3, H1(Ci)(Ki||j) + ci will be the child private key ski,j. 3) Child public key derivation: The procedure for determining a child's public key is abbreviated as Cpki,j=DeriveCpk(Ki,Ci,j). The child public key is calculated as Cpki,j = pki,j = KiH + Ci when H = H3, H1(Ci)(Ki||j).

B. Dynamic Accumulators

An element is created by adding a set S of values (x1, x2, x3...,xn) to show how the accumulator works. This is also the basis of our technology, which reduces the blockchain's storage overhead. The method in [35] is used to initialize the accumulator. For

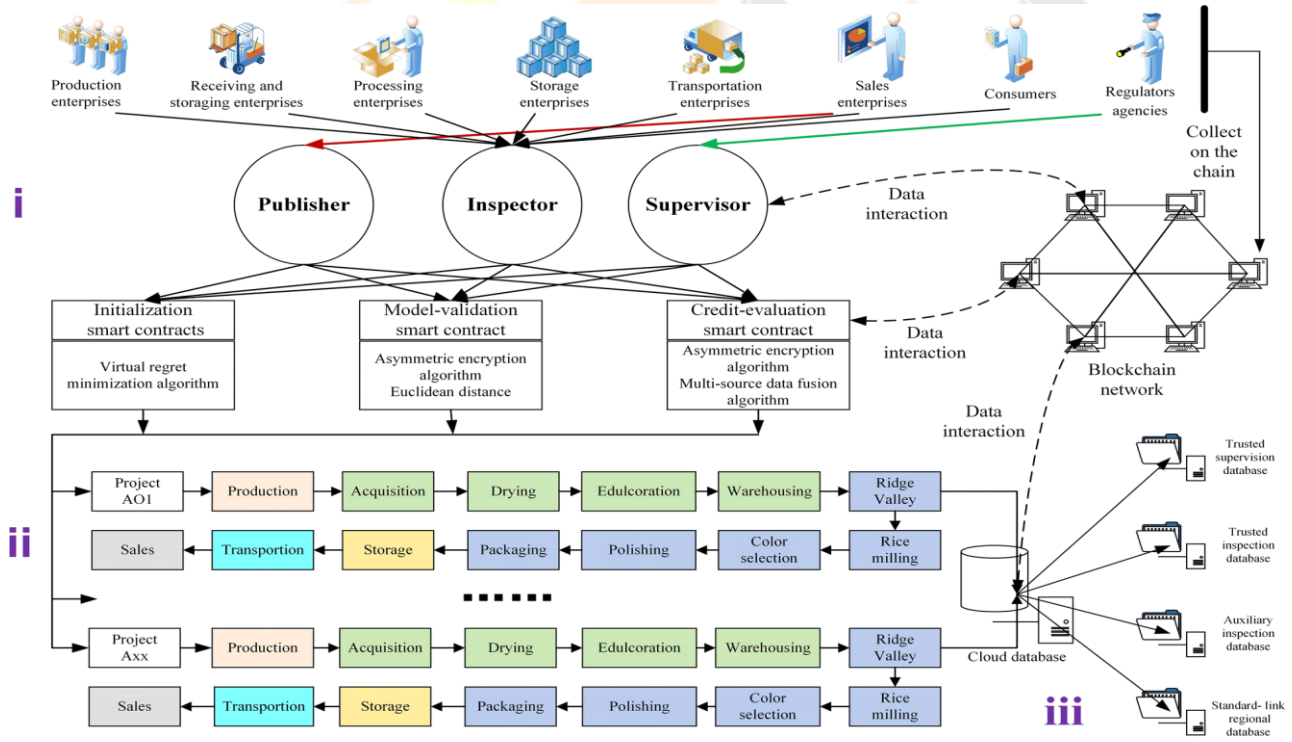

Fig. 1. System model.

Initially, a Type 3 bilinear group (G1, G2, GT,r, e) is selected as the accumulator. The bilinear mapping function is represented by the symbol He, and the cyclic groups G1, G2, and GT are all of prime order r. The generators of G1 and G2 are thus g1, g2, and y Z r is selected to compute Y = gy 2, Y = gy 1. Additionally, the element is computed as gQn 1 (y+xi) using g G1 and the value set S. For any value xi, there exists a witness Wi = 1/(y+xi).

If the value xi is added to the accumulator, the formula e(, g2) = e(Wi, Y gxi 2) will hold. Our protocol makes advantage of this novel feature to validate the unlinkable IDs of cross-domain IIoT devices.

The arbitrary number rw is a part of When Zr is initially chosen, three components are created: When combined with the prover's value xi, X = (Y)rw, K = H4(rw||xi) xi, and R = gH4(rw||xi) 2a are obtained by the prover. The verifier thus just has to verify that the equation e(W i,XgK 2a) ==? e(, R) using Wi, X, and K is correct, without having to ascertain the exact value of xi.

C. Smart contracts and blockchain technology

In its most basic form, the blockchain is a decentralized distributed ledger with highly accessible, impenetrable data that is synchronized across peer nodes through transactions [36]. The peer node will keep a copy of the digital ledger and build a state key-value database from the recorded transactions by carrying out the smart contract. The domain information is the value of this paper's key, which is each domain's blockchain address. The blockchain might successfully establish trust between many domains in this way.

IV. MODEL OF SYSTEM

A. MODEL OF SYSTEM

The blockchain and a number of different domains make up the system model.

- Trusted authority (TA): All entities within each domain trust the TA, which is responsible for registering IIoT devices and the domain server. By turning on the smart contract, the TA also has control over the data in the on-chain domain.
- IIoT device: IIoT devices are deployed in a single, designated IIoT domain in order to finish manufacturing tasks or connect with customers. IIoT devices in this research will offer a multitude of factors, be able to handle elliptic curve cryptography, and be quite heavy. They might communicate with the intra-domain server or link to organizations in other domains to create cross-domain partnerships.
- Server: A variety of services, such as data collecting, real-time access, and data analysis, can be provided by each domain's server. If the server needs data from other domains, it may send a request to the blockchain ledger.
- Blockchain: The blockchain acts as a public ledger to manage domain information. The TA and the server of each domain join the blockchain in order to register, update, query, or cancel the domain information by executing or querying the smart contract.. There are hence two prerequisites for the blockchain: first, the platform must be reliable and safe, and second, smart contracts must be supported. Based on these needs, integrating existing blockchain platforms into our system is quite practical and effective.

B. Threat Model

The following is a definition of the adversary's capabilities in our system:

(1) The adversary is assumed to have total control over the public channel in the Dolev-Yao (DY) paradigm [37]. The communications sent via the public channel were susceptible to interception, replay, eavesdropping, and modification by the attacker.

(2) IIoT devices would be vulnerable to cloning and physical assaults, such as power analysis. Thus, the keys and secrets stored in IIoT devices might be accessed by the attacker.

(3) In order to obtain sensitive information, such factors, the server is represented as a semi-honest entity that complies with protocol. However, contrary to what is implied in [38], the adversary will not be able to acquire the server's long-term secret keys. Every domain's TA is always reliable and safe.

(4) The recorded transactions remain unaltered and always accessible. The opponent can search the ledger, but they are unable to make transactions or interfere with the blockchain system.

The attacks listed below are those that are most likely to be carried out by the foe with the aforementioned skills:

(1) Impersonation attack: Using derived factors, the attacker would attempt to pass as the server or fabricate IIoT devices.

(2) Replay attack: To execute this attack, the enemy may intercept messages and play them back at a later time.

(3) Physical attack and factor loss attack: The adversary would take control of IIoT devices in order to carry out physical assaults and take device factors or secret keys.

(4) The desynchronization attack: The attacker may intercept the communication channel in order to desynchronize the update of important materials or identity information.

C. Security Assumptions

The following is a discussion of the security presumptions of our suggested protocol. Our system makes the assumption that Both the blockchain and the PUF circuit are secure. The fundamental security hypotheses and challenging difficulties are defined as follows as well:

The opponent cannot obtain s with a non-negligible probability given any P, sP G, according to

Definition 1 (Discrete Logarithm Problem for Elliptic Curve).

Definition 2: To compute abP given any aP, bP G (Computational Diffie-Hellman Problem for Elliptic Curves). There is a non-zero probability that the PPT opponent can solve the ECDH problem.

Definition 3 (q-Strong Diffie-Hellman Assumption): Given a Z p, a prime order p cyclic group G, a Z p generator g, and q > 0, the following function for any PPT algorithm A is deemed insignificant: PR [(x, g1/(a+x)) A (g, ga, ga2...,gaq)x Zp]

V. OUR SUGGESTED GUIDELINE

As seen in Fig. 3, this protocol is divided into four primary phases: registration, intradomain authentication ,cross-domain authentication ,and key negotiation.

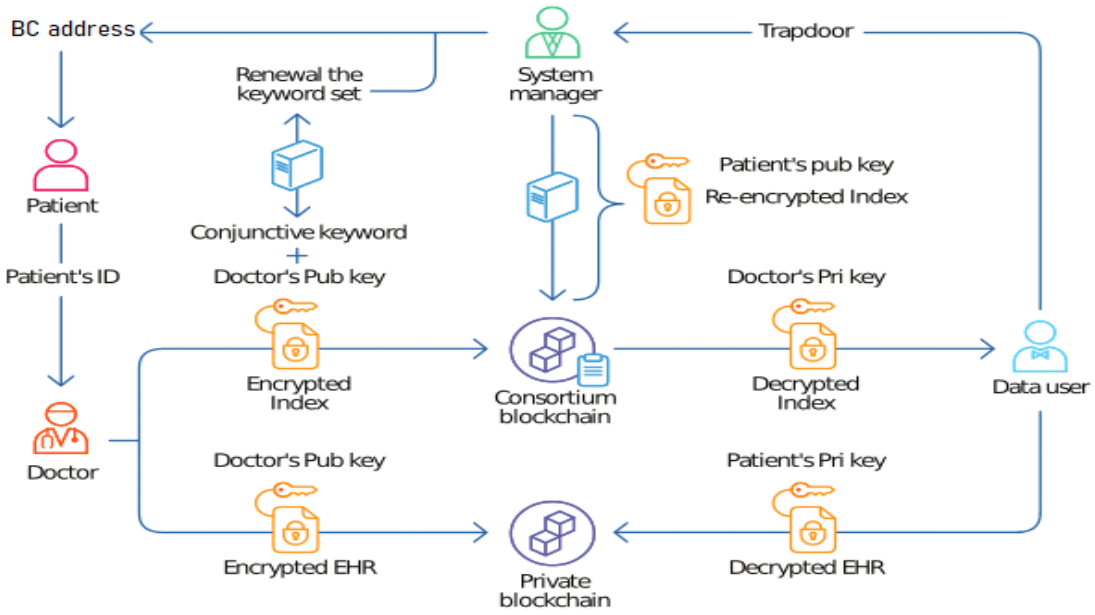

Fig. 2. Flow chart of our protocol.

A. Registration Phase

Each domain's parameters are initialized at this step. The server and other devices are then registered by the trustworthy authority. Finally, we talk about the revocation and update procedures.

1) Set the starting configuration of the system (R.1): The TA initially initializes the dynamic accumulator of each domain in order to acquire the tuple (Ga1, Ga2, Gat, r, e, y, Y, Y) mentioned in section III-B. Next, another elliptic curve cyclic group Ga of order r is chosen by the generator Pa. Furthermore, the following four hash functions are defined: H1: 0,1 0,1l1, H2: 0,1 0,1l2, H3: 0,1 0,1k Z r, and H4: 0,1 Z r, where k is the length of the key in hashes and l1 and l2 are the lengths of the hash functions. Lastly, the public parameters PPa (Ga1, Ga2, Gat, Ga, Papa, ga1, ga2, r, e, y, Y, Y, Hi1,2,3,4) and the domain identity Dida are made available.

2) Include the domain name in the server registration (R.2): The TA would disseminate the public/private key pair (sksa,PKsa) for the server. The TA then triggers the smart contract to upload the domain information (H1(Dida),PPa,version,a,PKsa) to the blockchain ledger. The entities of each domain will update the key-value state database with newly recorded transactions.

3) To create the PUF seed K (Root, I) with the selected random integer ri, the PUF response R is obtained. Without the PUF response, the attacker is still unable to determine the right parent secret key (ki,ci), even if factors are made known. Then, a list of fake IDs named Pid = (pid1, pid2, pidn) will be formed, along with further details about the components Ki and Ci. The device sends the (Ki,Ci,j,Tidi,j,Pid) to the TA..

4) Distribute crucial materials: After getting the IIoT device's registration request, the TA moves the components Ki,Ci to a selected value/witness pair xi/Wi that is already in the accumulator. After that, the mapping table MT = "Tidi,j: j,xi,Wi,Ki,Ci,Pid" will be created. Once the mapping table has been submitted to the server, the value xi is then returned to the device. The IIoT device saves the accumulator value xi locally as I: Tidi,j,j,ri,PUF,Pid after encrypting it into the element = EnH1(ki)(xi). The server uses its long-term secret key to safeguard the mapping table MT, much as the device-side does with the element xi.

5. Remove Xi from the accumulator, compute new = (1/y+xj), and activate the smart contract to publish the modified domain information to the blockchain in order to revoke and update device settings. Additionally, the server will update each (Xi, Wi) pair by calculating Wnew i = (Wi/new)1/(xjxi) and delete the information that has been revoked from the mapping database. If the device value Xi has to be changed, the TA and server will first revoke the old value xj as previously mentioned before allocating an unused new (xnew i,Wnew I) pair for the child public key pair. The server and device will then get the new values, xnew I and (xnew I,Wnew I), respectively.

B. Authentication within a domain

At this stage, IIoT devices configure mutual intra-domain authentication with the intra-domain server.

1. The module-operating device reads the Tidi,j,j,ri from storage before signing the intra-domain request. The accumulator value xi = DeH1(ki)() is then produced by deriving the jth child secret key ski,j using the parent private keys (ki,ci). to calculate D1 = n1PKsa and carry out the scent-out communications. In the next topics, we will not go into detail about TSi and Zi. The message M1 = "Tidi,j,j, N1, s, TS1, Z1" is sent to the server.

2. Verify the autograph (A.2). The Servera creates the child public key PKi,j=DeriveCpk(Ki,Ci,j) from the components (Ki,Ci,j) in accordance with Tidi,j in order to validate the signature. If the signature is authentic, the server accepts the device. Next, Tidi,j+1 = H1(j + 1||xi||N1||N2||N1) with N1, N2 is used to update Tidi,j and the key pair number. For example, SID = H1(Tidi,j||N1||N2) and H1(SID||D1||D2||xi) might be used to calculate the session key and identity if needed, where D2 = n2N1. The message M2 = N2, TS2, Z2 that the Servera transmits to the device a contains the secret element D1 = N1sksa. Signature of Schnorr [39] The next set of random numbers is n1, N1 = n1P, and s = skiH4(xi||N1) + n1.

3. Create a session key and authenticate the server (A.3). The device acknowledges the server after verifying the timestamp TS2 and hash value Z2. After changing Tidi, j, and j as previously mentioned, the session identity and session key are then computed to secure the next interactions.

C. Cross-domain Authentication

At this stage, the IIoT device must first authenticate with the intradomain server in order to get authorization for cross-domain requests. The server checks to see if the requestor's value is present in the accumulator by querying the blockchain ledger's accumulator.

1) Sign the request for a cross-domain (C.1): Step A.1 and Step C.1 are quite similar. The only change is that the identity IDb of the target deviceb in domain b, which has been encrypted into IDb, is now contained in the message M3 = "Tidi, Jj, N3, s, TS3, IDb, Z3".

2) Requesting authorization and signature confirmation (C.2-C3): After confirming the signature provided in step A.2, the Servera obtains IDb = DeH1(D 3||xi) (ID b) in step C.2. In step C.3, the Servera looks for domain b's details, such as PPb, ACCb, version, and PKsb, in the blockchain ledger. Using the formula D4, b = n4PKsb, four components—IDb and Ida, the witness Wi, and the domain information version—are encrypted into W i. Ultimately, the Servera grants permission for the cross-domain request and sends the message M4 = N4, b, TS4, W I, Z4 back to the device.

Remark: In this case, the IDs Ida and IDb are formed by combining the working mode with the IIoT device's true identity. 3) Cross-domain request forwarding and server verification (C.4–C5): After receiving M4, the device checks the date and hash value and modifies the Tidi,j as mentioned in step A before deciding whether to accept the server.

3. Step C.5 is where the ZKPoK method is constructed. The elements rw Z r, the accumulator value xi, the random number random, and K = H4(rw||xi) xi are combined to generate the values n5 = H4(random||xi), N5 = n5Pa, x = (Y)rw, and R = gH4(rw||xi) 2a. Device a then forwards the cross-domain request M5 = X, K, R, N4, b, N5, TS5, W I, Z5 to serverb.

4. (C.6–C.8) Verify the cross-domain request and provide the response: The server initially queries the blockchain ledger to obtain domain information (PPa, version, ACC, and PKs) when it receives M5.The serverb then decrypts W I using the algorithm DeH1(N4,bsksb)(W I) to retrieve the witness Wi of xi and the real identities IDb, Ida. Next, the server ascertains if e (Wi, XgK 2a) ==? e(ACCa,R). If this equation holds true, the accumulator considers the value xi of device A when it is in working mode I. The device and serverb have established a successful authentication. The device will be recognized by the server in step C.7, after which it will supply element N5 and the cross-domain authentication outcome.

D. Key negotiation and mutual cross-domain authentication

1) Exchange of Information inside a Domain: Mutual authentication is an intra-domain protocol. Additionally, the unilateral authentication technique covered in Section V-C involves deviceb authenticating device a. The device only has to repeat steps C.1–C.8 when mutual cross-domain authentication is required. Devica would then use deviceb to confirm the device's legitimacy.

2) The communicators on both sides will negotiate the session key and the session identity in order to protect the public channel. For the cross-domain request, we extended the Diffie-Hellman (ECDHE)-based Ephemeral Elliptic Curve-based key exchange method with a long-term secret xi. The communicators (devices a and b) will store the random numbers N5a = n5aP and N5b = n5bP. It would then be possible to calculate the session key, SK = H1(n5aN5b) = H1(n5bN5a), and the session identity, SID = H1(N5a||N5b). For intra-domain authentication, the process of negotiating the session key H1 (SID||D1||D2||xi) and the session identity SID = H1 (Tidi,j||N1||N2||D1) has already been described.

VI. EXAMINING SECURITY

This section initially uses BAN Logic to perform the formal security proof for our intra-domain and cross-domain authentication methods. The topic of security features and capabilities is then discussed.

A. BAN Logic's Formal Security Proof

To guarantee that communication is limited to authorized parties and that communicators trust one another, authentication is used. Additionally, the adversary is unable to access sensitive data or engage in hostile impersonation. Formally proving the security of the authentication becomes more challenging as protocols become more complicated. Nonetheless, the concept and rationale for authenticating security have recently been articulated through the use of the formal proof approach known as BAN Logic, which was first presented by Burrows, Abadi, and Needham (BAN) [40]. Furthermore, the evaluation [27] took into account BAN Logic as a formal security proving method for blockchain-based systems. This work employs BAN logic to offer formal security proof.

To provide a clearer understanding of the formal proof of the BAN logic, we first supply the necessary definitions and notations.

(1) P|≡ M. The entity P accepts the information M.

(2) P CM. The message M is seen by the entity P.

(3) P|∼ M. The message M was sent by the entity P.

(4) P|⇒ X. Fully in charge of the message X is the entity P.

(5) #(M). It is a new message, M.

(6) {M}K. M is a message that K has encrypted.

(7) < M >Y. Together with Y, the message M is sent.

(8) P SK ←−−→ Q. The secret SK is shared by entities P and Q.

(9) P|≡ KQ −−→ Q. P has faith in Q's public key.

The rules that govern the proving process are then described.

R.1 The message-meaning rule states that P will presume M was sent by Q if P thinks P

and Q share K and P receives M that K has encrypted.

$$Q, P \text{ CMK} \leftarrow\!\!-\!\!-\!\!\rightarrow P|\equiv P \text{ SK } P|\equiv Q|\sim M$$

In addition, we incorporate the following accumulator rule and message-signature rule into the message-meaning rule:

Rule 2: Message signature: If a message is signed with Q's private key using a secure signing method and P believes Q's public key, P will believe that M was sent by Q.

$$P \text{ C}\{M\}K-1 \text{ Q}, P|\equiv KQ \text{ −−→ Q } P|\equiv Q|\sim M$$

R.3 Accumulator Rule: P will acknowledge that Q delivered message M if P accepts witness Wi of xi and accumulator ACC of xi, and P receives message M with value xi of Q.

$$QC < M > xi, P|\equiv ACC,Wi \text{ −−−−→→ Q, } P|\equiv Q|\sim M$$

R.4 Rule for once-through verification: It confirms the message's timeliness. P|TM

#(M), P|TM Q|TM M P|TM Q|TM M

R.5 Rule of Jurisdiction: P believes X if P believes Q fully controls and believes that X.

$$Q|\equiv X, P|\geq Q|\geq X \; P|\geq X$$

R.6 Belief Rule: If P thinks Q trusts the communication (X, Y), then P believes Q trusts X.

$$P|TMQ|TM (X, Y) \; P|TMQ|TM X$$

B. official documentation of intra-domain authentication

We evaluate our technique in light of the proof procedures covered in work [42].

B.1 Authentication's purpose:

The aims of our suggested intra-domain authentication should be as follows. Q is the server, while P is the IIoT device. Goals 1 and 2 are P| (P SK Q) and Q| (P SK Q), respectively.

B.2 The process of idealization the idealized procedure for intra-domain authentication is: (1) The message M1 is as follows: P Q:

(Tidi, j, j, N1, s, TS1, Z1, xi, D1(PKsa)1).

(2) The text in the message M2 is Q P: (Z2, N1, N2, TS2 > (D1, xi)).

(3) The syllables P and Q are written as follows: (Tidi, j, N1, N2 > (D1, xi)).

The communication procedure that will follow is idealized in this message.

B.3 The process of assumption:

First assumptions are established as follows: S2: P | P xi; S1: P xi | Q | P xi Q Q; S3: Q| # (N1, TS1); S4: P| # (N2, TS2); Q S5: "Q| PKi,j" P. This assumption shows that the kid public key of the IIoT device is on the server. S6: SK = Q|≡ P|. S7: SK. B. = P|≡ Q|.4 Technique of demonstration We now use the guidelines and suppositions to perform the BAN logic proof.

(1) For message M1, Q notes that (Tidi, j, j, N1, s, TS1, Z1, xi, D1 >(PKsa)1). By considering the message-signature rule (R.2), the security of the Schnorr signature as shown in work [39], and the assumption S5, we arrive at step (2).

(2) Q|≡ P|~ M1. Using the nonce-verification rule to step 2, we obtain Q|P|M1. By utilizing its private key, sksa, the Servera may get D1 = sksaN1. Next, we derive step (3) using S1 and the belief rule (R.6). (D1, N1, xi) Q|P|.

(3) For the message Mc, Q notes Tidi, j, N1, N2 > (D1, xi). Step (3)'s evidence is combined with the message-meaning rule to construct step (4).

(4) Q|≡ P|~ Mc. By combining step (4) with the nonce-verification rule, we are able to produce step (5).

5. (N1,N2,D1,xi) Q| P|, We examine the proofs obtained in stages (3), (5), D2 = n1N2, S1, and the session key, SK = H1.From (SID||D1||D2||xi), we deduce that Q| P| P SK Q. The justification rule with the assumption S6 leads us to the conclusion that Q| P SK Q (Goal 2).

(6) Only Q is able to obtain D1 for the message M2 since the ECDH problem remains unresolved. Step (7) can be reached by applying assumption S2 to the message-meaning rule (R.1).

(7) P| ≡ Q| ~ M2. By adding the S4 to the nonce verification rule, we are able to get P|Q|M2. Next, the belief rule is used to obtain.

(8) (N1, N2, P|Q| (8), D1, D2). from the S2 and step (8). Based on the session key SK = H1(SID||D1||D2||xi) and the SID, we conclude that P| Q| P SK Q. By applying the justification rule to premise S7, we may conclude that P| P SK Q (Goal 1).

C. Official documentation proving cross-domain identity

C.1 Authentication's purpose:

The following should be the goals of our recommended cross-domain authentication. P is the requestor device (device a), and Q is the receiver Serverb. It should be remembered that Serverb can send the outcomes securely to deviceb. For ease of comprehension, the server is used in this instance in place of the device. P| (P N5a Q) and Q| (P N5b Q) are goals 3 and 4, respectively. Accepting each other's authentication as well as the random numbers N5a and N5b is the aim of cross-domain authentication. Using the random numbers, the session key SK = n5aN5b = n5bN5a between device a and device b based on ECDHE will be formed.

C.2 Process of Idealization:

1) The message from M5a: P Q: (X, K, R >xi, N4, N5a, W, i >D4) The device a sends serverb the M5a cross-domain request that is allowed by servera.

C.3 The assumption procedure is as follows:

S1: Q| Accie P. This assumption simulates server B requesting the domain-specific accumulator Accie by contacting the blockchain ledger.

Q|=# (N4, N5a, TS5) in S2; Q|=P|==Na in S3

C.4 Method of proof:

(1) For message M5a, Q notes that (X, K, R >xi, N4, N5a, Wi >D4). First, D4 = sksbN4 is computed using the private key sksb of serverb in order to validate the cross-domain request M5a and retrieve the real witness Wi of xi. The next step is to determine if xi is a component of Accie using S1 and the equation e (Wi, XgK 2a) ==? e(ACCi,R). If so, the accumulator rule (R.3) might be used to produce step (2).

(2) Q| ≡ P| ~ M5a. By using the nonce-verification rule on step (2) and the S2, we are able to acquire Q| P| M5a. Then, step (3) is obtained by using the belief rule.

Q | P | (P N5a Q)

(3). Combining S3 with the justification rule leads to step (4).

(4) Q|P N5a Q. The server would provide the N5ato device and the outcome of the cross-domain authentication to the client. Similarly, we obtain P| P N5b Q (objective 4) if mutual authentication is utilized.

The goals of cross-domain authentication would be achieved. Thus, devices a and b in different domains might authenticate each other. The random numbers N5a and N5b will also be used to negotiate the session key, SK = H1(n5bN5a) = H1(n5aN5b), in order to secure the cross-domain public channel.

D. Talking About Security Features and Capabilities

We start out by talking about our protocol's security characteristics.

(1) Mutual authentication inside the same domain between a device and an IIoT server.

a) To authenticate the IIoT device, the server checks the signatures. If an attacker is able to successfully construct a genuine message M1 to circumvent the server's authentication, we can acquire sP = PKi, jH4(xi||N1) + N1. To provide the same unpredictability, the same random tape is used. But in order to determine the correct login message, a separate hash oracle output is used. Consequently, the simulator may arrive at (s s) (H4(xi||N1) H4(xi||N1))1 as a solution for a given instance (P,ski,jP), which would be inconsistent with the reported difficulty level of the ECDL issue. Consequently, the attacker's attempt to authenticate using a forged IIoT device is unsuccessful.

b) The Z2 hash value is used to validate the server. Assume that the server's private key is kept a secret and that a hacker fabricates a legitimate message M2 = N2, TS2, Z2 = H2(D1||) to go past the device's verification. It shows that the attacker used the hash oracle to construct D 1 = n1PKs = sksn1P. Di may be the solution to the (P, sksP, n1P) problem, which questions the ECDH problem's level of complexity. Consequently, no hacker could fool the verification by impersonating the service.

(2) Mutual authentication across domains in the Internet of Things The IIoT device is validated by the on-chain accumulator Accie. The server must first decode the witness Wi before obtaining it. Next, the accumulator ACCi of the blockchain is examined using our ZKPoK method to determine if the value xi of the device an is contained in ACCi. It is challenging for the enemy to access the right xi since it is both securely secured at the device's side and concealed in X, R. In order to pass the verification, Against the q-SDH assumption, the adversary can only provide a new valid value/witness pair. As a result, unilateral cross-domain authentication

is guaranteed. Similarly, the deviceb might also create cross-domain mutual authentication by performing mirror activities.

(3) Anonymity and unlink ability: In our protocols,the IIoT devices' identities are anonymous. IIoT devices only transmit temporary pseudo IDs Tidi,j or genuine identities encoded within IDb and Wi. Since their genuine identities won't be revealed, IIoT devices' privacy may be guaranteed. To ensure unlinkability, just the domain information will be extracted from the blockchain ledger; no public keys of the device will be accessible during the authentication procedure. Furthermore, Taij's false identity is frequently connected to and modified. It is difficult for the attacker to link two messages to the same requestor device. In general, our technique may preserve privacy while achieving unlikability and anonymity.

(4) To ensure the security of multiple factors, the multiple factors are transformed into the parent secret keys, ki and ci, which are then saved on the server as Ki = Kip and Ci = ciP. The opponent is unable to solve the ECDL issue in order to get parent secret keys and device factors. Consequently, it would be possible to ensure the different pieces' server-side protection.

(5) Our protocols' session keys are calculated as follows: SK = H1(SID||D1||D2||xi) and SK = H1(n5aN5b), respectively. Assume that the historical messages and secret keys (sksa, sksb, and xi) are released to the public. Without the correct random numbers, it is still difficult for the attacker to compromise the elements D2 = n1N2 and n5 = H4(random||xi) and figure out the correct session key. Thus, our protocol may achieve great forward secrecy.

The security features are then shown to demonstrate how our protocol can fend off several potential known assaults.

(1) Defense against impersonation attacks: If an attack attempts to mimic an IIoT device or server that has been authorized, it must violate our mutual authentication procedures. Our mutual authentication systems' security has all been shown, nevertheless. Attack through impersonation will not be successful.

(2) Replay attack resistance is provided by the random number approach used in our protocol. Since the random integers N1 through N5 are distinct for every session, the attacker cannot effectively complete the verification by repeating requests that are lawfully permitted. Consequently, our protocol is resilient to a replay attack.

(3) It is thought that the PUF circuit is safe as the attacker cannot duplicate or anticipate the right PUF response R to calculate the Pulsed H1(R||ri). Both loss of components and resistance to physical attack. If factors of the corrupted IIoT device are exposed, the attacker still cannot accurately encode these parameters to determine the appropriate ci without the PUF seed. As a result, our technique might thwart an assault involving the loss of factors.

(4) Defense against desynchronization assault: This attack will disrupt the channel and force the communicators to only update the Tidi,j on one side. However, as explained in [41], the list of fictitious identities Pid is used as a defense against this kind of attack. In case of a desynchronization, any of the idle pidi might replace the Tidi,j and bring the system back into synchronization.

VII. EVALUATION OF PERFORMANCE

We conducted the experiment to evaluate the efficacy of our methods. The results of six critical metrics—efficiency, computational and communication overhead, on-chain storage overhead, smart contract performance, and functional comparability—are covered in this section.

A. Conditions for Experiments

1) IIoT domain entities: We created two IIoT domains in simulation for our experiment. Every domain has an IoT device, a server, and a trustworthy authority. Servers and trustworthy authorities in two domains were deployed on a laptop equipped with an Intel Core i510210U processor operating at 1.6 GHz and 16 GB of RAM. The IIoT devices were run by two Raspberry Pi 3B+ computers.

2) Blockchain network: Hyperledger Fabric, an open-source project, integrated both the blockchain network and the smart contract. Both servers and trustworthy authorities have installed blockchain clients developed by the Java SDK. To control the blockchain ledger, they might call upon or inquire about the smart contract.

3) IIoT devices were equipped with three distinct types of factors: the PUF key, the biometric key, and the serial number. The DRAM PUF that we established in our previous work [43] provided the 128-bit PUF response in our experiment. Additionally, the widely used facial feature extraction project, Facial Recognition, was used to determine the user's biometric information. Next, a fuzzy extractor was developed to transform the PUF response and biometric feature into a 128-bit PUF key and a biometric key, respectively, using the BCH error correction code. Then, and only then, was the serial number loaded from the device storage.

4) Algorithms: Our proposed protocols were implemented at the application layer using the HTTP protocol, which may be readily converted to CoAP in IIoT applications. Cryptographic algorithms were developed using the Java JPBC 2.0.0 library and Bouncy Castle 1.60. It should be noted that the Secp256R1 elliptic curve and Type pairing were applied in our implementation.

The average costs of all the cryptographic operations used in our protocols are shown in Table II, where H is the general hash function. The exponentiations in Gr1/G1/G2 are represented by Exp1, Exp2, and Ext, and the scalar multiplications by SMr1, SM1, and SM2. MT is the multiplication in GT, G1/G2/GT, PM2 and PMr1, and BP is the bilinear pairing. In G2/Gr1, the point increases are PM2 and PMr1.

TABLE II

CRYPTOGRAPHY OPERATIONS' AVERAGE COSTS (UNIT: MILLISECOND)

Notes	Gadget	Waiter	Notes	Gadget	Waiter
H	1.02	0.01	Exp1	51.62	5.52
SMr1	27.45	1.53	Exp2	105.86	8.92
SM1	56.89	5.87	ExpT	478.13	31.24
SM2	114.92	9.26	PM2	0.38	0.03
MT	4.13	0.14	BP	2168.47	132.94
PAr1	0.10	0.01			

The results of a series of authentication requests are displayed in Figure 7. The PUF key and the serial number are the factors used by all devices; only the requestor (device) of the cross-domain authentication requires an extra biometric key.

B. Efficiency

For both intra-domain and cross-domain mutual authentication, 1000 distinct consecutive requests were submitted. Every request has its real time cost disclosed, with the average time being indicated in red. It takes 1450.36 Ms to process the unilateral cross-domain request, which already accounts for the time needed to extract the biometric characteristic at the requestor's end (device a). In just 319.22 Ms, the intra-domain

request is completed. It takes around 2577.28 MS to create the mutual cross-domain authentication, in which the requestor and the receiver (devices a and b) authenticate one another and establish the session key. Furthermore, the real time of each request is reliable and varies within a red dashed line representing the average execution time. To sum up, our protocols work well and consistently for IIoT applications.

TABLE III

Data about lengthy cryptography processes (key negotiation; CDA: unidirectional cross-domain authentication; intra-domain authentication; IDA: IDA)

Organization	IDA and KN	CDA
devicea	5SMr1	4SMr1+2Exp2
servera	6SMr1+2PAr1	6SMr1+2PAr1
deviceb	5SMr1	/
serverb	6SMr1+2PAr1	SMr1+Exp2+PM2+2BP

TABLE IV

COMPARISON OF THE COMPUTATION OVERHEAD FOR THE DUAL CDA AND KN DEVICES

Schemes	Charges for the Subject Device	Expenses for the Object Device
Block CAM [10]	4SMr1+PAr1	4SMr1+PAr1
SCCA [12]	3SMr1	3SMr1
BASDA [4]	2SMr1+SM1+ExpT	2SMr1+SM1+ExpT
Our	6SMr1 +2Exp2	6SMr1 +2Exp2

The simulation findings for the processing costs incurred by IIoT devices during key negotiation (KN) and mutual cross-domain authentication (CDA) are displayed in Figure 8.

C. Overhead in Computation

To compute the computation overhead, we first tally the most time-consuming activities. Then, in order to compare our technique with relevant research, we provided simulation findings.

1) Theoretical analysis: Table III lists the processes that need the greatest amount of time for every entity. Establishing intra-domain authentication and key negotiation requires the server 6SMr1+2PAr1 and the IIoT device 5SMr1. The cost of completing the unilateral cross-domain authentication request is 4SMr1+2Exp2 for the subject device and 6SMr1+2PAr1 for the server. Additionally, when the servrb 3SMr1+Exp2+PM2+PAr1+2BP has validated the request, the object deviceb does nothing except receive the result and the random number N5. The overhead for requests for mutual cross-domain is easy to compute, as unilateral cross-domain is a symmetrical process.

2) Results of the simulation: We contrast our protocol's computational burden on IIoT devices with that of existing blockchain-based cross-domain authentication systems, including Block CAM [10], SCCA [12], and BASDA [4]. We do not compare server expenses because it is presumed that the server has adequate computational capacity. Table IV shows that throughout the symmetric cross domain authentication

procedure, There is no intensive operation required, and the expenses of the subject device (device a) and the object device (device b) are equal. such as bilinear pairing, is carried out by the devices. Additionally, Fig. 8 displays the simulation outcomes computed using the same cryptography parameters. It demonstrates that the mutual CDA and KN mechanisms have a total cost that is linear in the number of devices. Additionally, it is simple to observe that our protocol outperforms BASDA [4] and that Block CAM and SCCA have computation costs that are similar to but less than our protocol. The combined results of our theoretical study and simulations indicate that the computing overhead of our protocol is reasonable and suitable for the IIoT devices included in our system model.

D. Communication Overhead

In this section, we assess the communication overhead. The length of the element in G/G1/G2 is 256 bits, 256 bits, and 1024 bits. The output of the H2 is 256 bits, as is the signature s. The element X is 128 bits. Tidi,j is 128 bits, whereas the genuine identity ID, TS, and j are all 32 bits. With these configurations, the device a transmits the cross-domain request M3 (156 bytes) to Servera, which returns the permitted request M4 (185 bytes). The value x1 is then encoded by the devciea into the 709-byte message M5. To establish cross-domain authentication, the message M5 will be transmitted to the cross-domain server. Both our protocol and BASDA [4] need the domain server to provide permission for all cross-domain requests. But the components of the other two techniques, Block CAM [10] and SCCA [12], may transmit cross-domain requests directly without requesting authorizations, which reduces communication costs. With a total of 768 bytes, the most relevant piece of work, BASDA [4], has lower transmission costs than our prototype. Our protocol's device should input the elements X, K, R, and W into the cross-domain request message M5, which is used to create the accumulator's ZKPoK method. which is the primary source of the difference gap

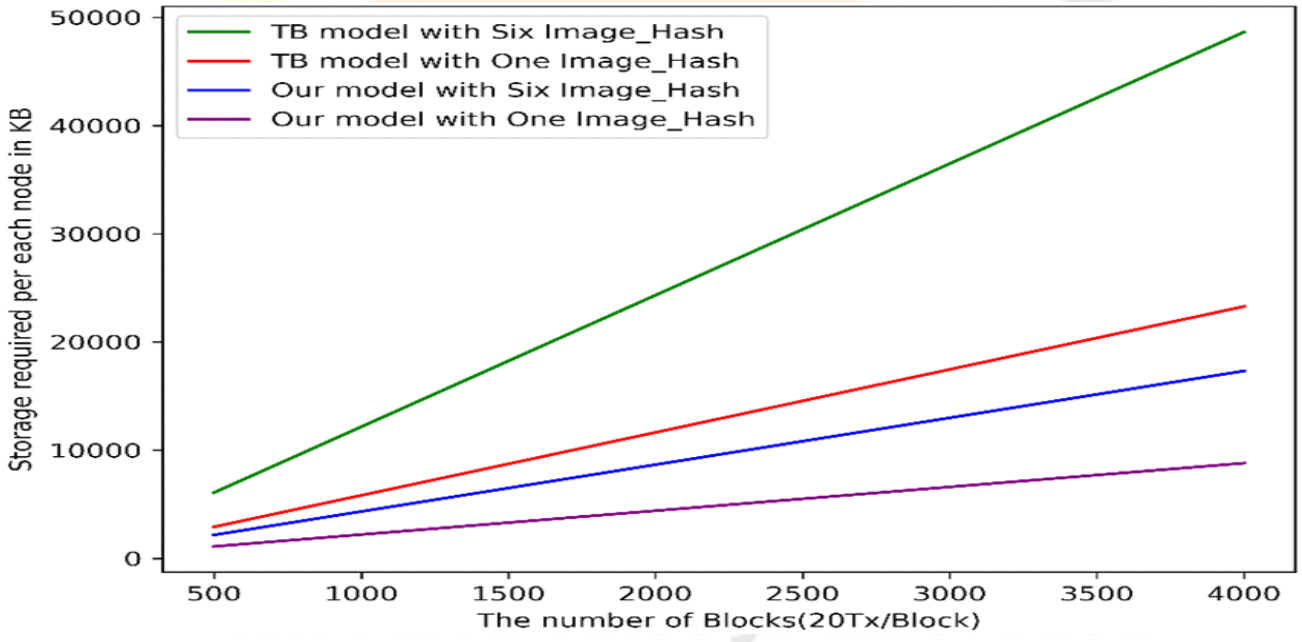

Figure 9. The blockchain's storage overhead.

E. Blockchain Storage Overhead

To demonstrate the advantages of our protocol while accounting for the higher storage needs of the blockchain, we created 1000 public keys for each domain in the comparison. While our solution consumes 0.14 KB for each domain, the on-chain storage requirements for each domain for Block CAM [10], SCCA [12], and BASDA [4] are 39.06 KB, 31.25 KB, and 15.63 KB, respectively. Our process is shown in Figure 9. performs noticeably better than the other three methods as the number of domains rises. Our suggested trust building

approach might be used to illustrate our protocol's benefit. Instead of storing several public keys or certificates on the blockchain, this unique technique keeps the accumulator. In general, our approach actually significantly lowers the application protocol layer's on-chain storage cost.

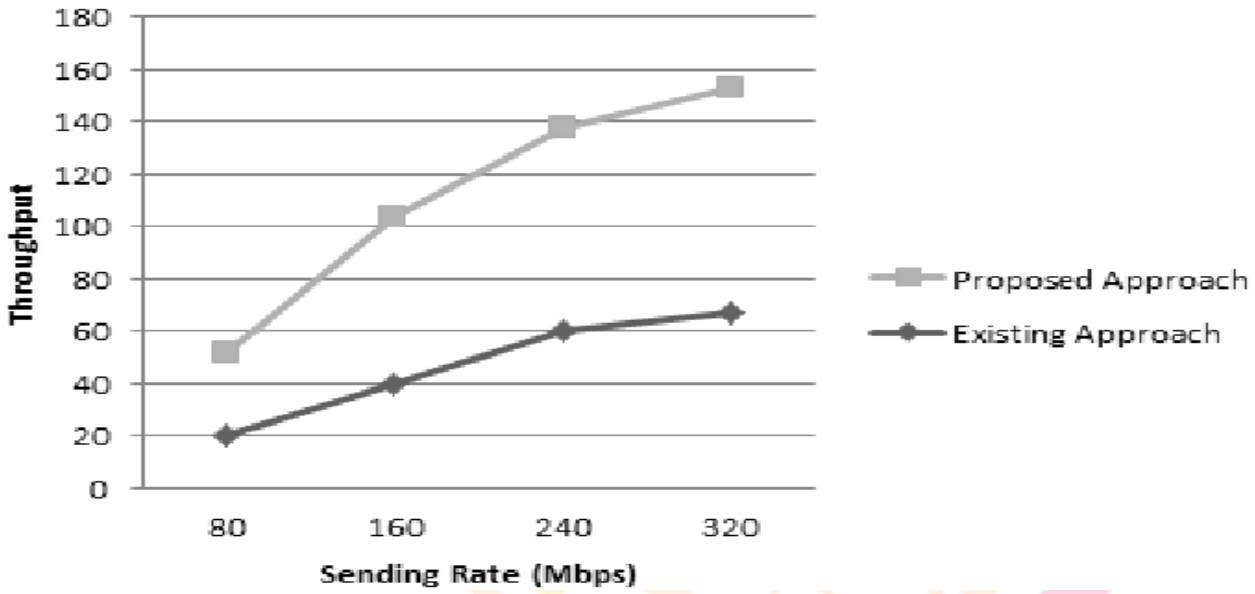

Figure 10 shows the outcomes of an experiment using a smart contract across many blockchain networks.

F. Assessment of Smart Contract Performance

In our method, the authentication procedure just asks questions of the smart contract. The smart contract will only be activated by completing the registration and revocation stages. We built blockchain networks with many domains to examine the smart contract performance in this part. Batch Timeout and Batch Size settings are set to 2 seconds and 10, respectively, to start the blockchain. Additionally, we set the desired maximum size at 512 KB and the absolute maximum block size at 99 MB.These core possibilities are quite comparable to work, and our consortium's blockchain has no transaction costs [4]. The primary distinction is that they used a Batch Timeout of 0.05 seconds in their study. We also performed simulations on five different kinds of networks with different numbers of IIoT domains.

We tracked the smart contract's query and invoked latency in order to assess performance. Only the smart contract, which is also intended to be a key-value state database, is accessed by the server using queries to retrieve information from the local blockchain ledger copy. There won't be any latency issues as a result of the concurrent requests. In our protocol, the query latency consistently maintains an effective value of 19.59 milliseconds. We recorded invocation delays and simulated concurrent transactions given at different transmission speeds using the open-source Hyperledger Caliper project. When the data is added to the blockchain and the smart contract is turned on, the latency is recoded. A batch timeout of two seconds. However, when more concurrent transactions must be handled, the invocation latency will grow. An intriguing phenomenon seems to be that with rising transmit rates, each type of network experiences a dramatic growth point. It suggests that the sharp point will arrive more quickly the more domains the network has. This result is to be expected given that sorting transactions and verifying endorsed proposals take more time on our instanced blockchain network with more domains. Our networks are limited to having a throughput of no more than 50 TPS. However, it won't have an impact on how effective authentication is as our protocol's authentication mechanism does not need the proposal of transactions. Additionally, when used in industrial settings, the blockchain network's throughput might exceed 3500 TPS or even 20000 TPS [44]. To ensure efficiency, the query delay is often satisfied. Furthermore, it would not be impacted by the efficacy of the

invocation latency performance, which might be further improved by blockchain network optimization, as was done in the study [44].

Table V

SUMMARY OF SECURITY FEATURES AND FUNCTIONALITY

Features and functionalities related to security	Block CAM [10]	SCCA [12]	BASDA [4]	Our protocol
Reciprocal verification	Yes	Yes	Yes	Yes
Privacy	Yes	Yes	Yes	Yes
Unlink ability	No	No	Yes	Yes
Defying an onslaught from factors lost	No	No	No	Yes
Defying replay assaults	Yes	Yes	Yes	Yes
Robust forward secrecy	No	Yes	Yes	Yes
Effectiveness	Yes	Yes	No	Yes
Minimal overhead for on-chain storage	No	No	No	Yes

This feature indicates that the blockchain system's throughput will not limit the authentication process's duration.

G. A Comparison of Functionality and Security Features

In this part, we compare the security features and purposes of our work with relevant protocols [4, 10, and 12]. Table V lists the parallels and differences in order to illustrate the novel security aspects that our method offers. Key security requirements that are addressed by contemporary protocols include mutual authentication, anonymity, and resilience to replay attacks, as shown in Table V. In addition, strong forward secrecy will be provided to maintain session keys in the event that secure channels need to be established, as mandated by [4], [12], and our work. Given that additional transactions must be suggested throughout the authentication process, Protocol [4] is not considered to be efficient. Only our approach and protocol are able to accomplish unlink ability [4]. This comparison demonstrates that only our protocol offers crucial characteristics like reduced on-chain storage costs and resilience to factor attacks.

VIII. CONCLUSION

We have developed an efficient and privacy-preserving multi-factor device authentication technique using blockchain to ensure cross-domain IIoT device collaboration. Formal security proof has been presented by BAN Logic, and the security discussion shows that our protocol not only provides resistance against the loss of factor attack but also provides protection for the multi-factor database. Furthermore, anonymity and unlikability are also ensured to safeguard privacy. Our protocols for intradomain, unilateral cross-domain, and mutual cross-domain take 319.22 MS, 1450.36 MS, and 2577.28 MS, respectively, and are reliable and

efficient. Additionally, each domain with 1000 public keys now has a 0.14 KB on-chain storage cost. The scalability of the smart contract is demonstrated by evaluating its performance.

REFERENCES

[1] "Industrial internet of things: Challenges, opportunities, and directions," IEEE Transactions on Industrial Informatics, vol. PP, no. 11, pp. 4724-4734, 2018. E. Sesani, A. Saifullah, S. Han, U. Jennesa, and M. Gidlund.

[2] "Edge computing in industrial internet of things: Architecture, advances and challenges," IEEE Communications Surveys and Tutorials, vol. PP, no. 99, pp. 1-1, 2020. T. Qiu, J. Chi, X. Zhou, Z. Ning, and D. O. Wu.

[3] "Internet of things in industries: A survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, 2014. L. D. Xu, W. He, and S. Li.

[4] "Blockchain-assisted secure device authentication for cross-domain industrial IoT," IEEE Journal on Selected Areas in Communications, vol. PP, no. 99, pp. 1–2, 2020.

[5] "Efficient multi-factor authenticated key exchange scheme for mobile communications," IEEE Transactions on Dependable and Secure Computing, pp. 1–1, R. Zhang, Y. Xiao, S. Sun, and H. Ma, 2017.

[6] "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," IEEE Internet of Things Journal, vol. 8, no. 2, 2021, pp. 844–859. Z. Li, Z. Yang, P. Zelechowski, and J. Zhou.

[7] "Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment," IEEE Transactions on Dependable and Secure Computing, vol. PP, no. 99, pp. 1-1, 2016. S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, and A. V. Vasilakos.

[8] J. Wang, L. Wu, K. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," IEEE Transactions on Industrial Informatics, vol. PP, no. 99, pages 1–2, 2019.

[9] "xdbauth: Blockchain based cross domain authentication and authorization framework for internet of things," G. Ali, N. Ahmad, Y. Cao, S. Khan, H. Cruickshank, E. A. Qazi, and A. Ali, IEEE Access, vol. 8, pp. 58800-58816, 2020.

[10] "Block cam: A blockchain-based cross domain authentication model," in Conference Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).

[11] "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1-1, 2017. L. Ao, H. Cruickshank, C. Yue, P. Asuquo, C. Ogah, and Z. Sun.

[12] "Smart contract-based cross domain authentication and key agreement system for heterogeneous wireless networks," in G. Li, Y. Wang, B. Zhang, and S. Lu, "Mobile Information Systems," vol. 2020, no. 29, pp. 1–16, 2020.

[13] "Btcas: A blockchain based thoroughly cross-domain authentication scheme," Journal of Information Security and Applications, vol. 55, no. 102538, 2020.

[14] "Bcppa: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–13, 2020.

[15] "Master-slave chain based trusted cross-domain authentication mechanism in iot," Journal of Network and Computer Applications, vol. 172, no. 102812, 2020. S. Guo, F. Wang, N. Zhang, F. Qi, and X. Qiu.

[16] "A blockchain-based mutual authentication scheme for collaborative edge computing," IEEE Transactions on Computational Social Systems, vol. PP, no. 99, pp. 1–13, 2021. G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin.

[17] USENIX 2015, "The Pythia prf service," 2015.

[18] "Examining smart-card security under the threat of power analysis attacks," IEEE Transactions on Computers, vol. 51, no. 5, pp. 541–552, 2002. T. S. Messages, E. A. Dabish, and R. H. Sloan.

[19] "A privacy-aware puts based multi-server authentication protocol in cloud-edge iot systems using blockchain," IEEE Internet of Things Journal, pp. 1-1, 2021. Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng.

[20] "A lightweight scalable protocol for public blockchain," Journal of Computer Research and Development, vol. 57, no. 7, 2020, pp. 1555–1567. [Online]. Accessible at ISI://CSCD:6759334.

[21] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, "A blockchain-based privacy-preserving authentication scheme for vanes," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. PP, no. 99, pages 1–10, 2019.

[22] "Privbiomtauth: Privacy Preserving Biometrics-Based and User-Centric Protocol for User Authentication from Mobile Phones," IEEE Transactions on Information Forensics and Security, vol. PP, no. 99, pp. 1-1, 2017.

[23] "Secure remote multi-factor authentication scheme based on chaotic map zero-knowledge proof for crowdsourcing internet of things," W. Liu, X. Wang, and W. Peng, IEEE Access, vol. PP, no. 99, pp. 1-1, 2019.

[24] "Blockchain for internet of things: A survey," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076-8094, 2019. H. N. Dai, Z. Zheng, and Y. Zhang.

[25] "Blockchain technologies for the internet of things: Research issues and challenges," IEEE Internet of Things Journal, vol. PP, no. 99, 2018. M. A. Farrag, M. Der dour, M. Mukherjee, A. Derh Ab, and H. Janicke.

[26] "Security challenges and opportunities for smart contracts on the internet of things: A survey," IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1–1, 2021. K. Peng, M. Li, H. Huang, C. Wang, and K. Choo.

[27] "The Performance Evaluation of Blockchain based Security and Privacy Systems for the Internet of Things: A Tutorial," M. A. Farrag and S. Lei, IEEE Internet of Things Journal, vol. PP, no. 99, 2021.

[28] "Master-slave blockchain based cross-domain trust access mechanism for up iot," in Conference Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS).

[29] "Toward cross-domain dynamic accumulator authentication based on blockchain in internet of things," IEEE Transactions on Industrial Informatics, vol. 18, no. 4, 2021, pp. 2858–2867.

[30] "Iot passport: A blockchain-based trust framework for collaborative internet-of-things," in the 24th ACM Symposium, Conference Proceedings, by B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu.

[31] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the iot scenario," IEEE Access, vol. PP, no. 99, 2019; pp. 1–2.

[32] "Ibra: An identity based cross-domain authentication scheme for the internet of things," Electronics, vol. 9, no. 4, p. 634, 2020. X. Jia, N. Hu, S. Su, S. Yin, and C. Zhang.

[33] "Bidm:a blockchain-enabled cross-domain identity management system," Journal of Communications and Information Networks, vol. 6, no. 1, p. 15, 2021. R. Chen, F. Shu, S. Huang, L. Huang, H. Liu, J. Liu, and K. Lei.

[34] "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4660–4670, 2019.

[35] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang. "Accumulators from bilinear pairings and applications," N. Lan, Springer, Berlin, Heidelberg, 2005.

[36]"Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," IEEE Transactions on Industrial Informatics, vol. PP, no. 99, pp. 1-1, 2019.

[37] D. Dolev and A. C. Yao, together with MEMBER, IEEE, "On the Security of Public Key Protocols," Information Theory IEEE Transactions on, vol. 29, no. 2, pp. 198-208, 1981.